Available online at www.sciencedirect.com
SCIENCE DIRECT®
Discrete Applied Mathematics 145 (2005) 210–218
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Tree-decompositions of small pathwidth☆

Jan Arne Telle

Department of Informatics, University of Bergen, N-5020 Bergen, Norway

Received 3 February 2002; received in revised form 11 March 2003; accepted 16 January 2004
Available online 5 October 2004

Abstract

Motivated by the desire to speed up dynamic programming algorithms for graphs of bounded treewidth, we initiate a study of the tradeoff between width and pathwidth of tree-decompositions. We therefore investigate the catwidth parameter $catw(G)$ which is the minimum width of any tree-decomposition (T, X) of a graph G when the pathwidth $pw(T)$ of the tree T is 1. The catwidth parameter lies between the treewidth and the pathwidth of the graph, $tw(G) \leqslant catw(G) \leqslant pw(G)$, and just as treewidth relates to chordal graphs and pathwidth relates to interval graphs, catwidth relates to what we call catval graphs. We introduce the notion of an extended asteroidal triple (XAT) and characterize catval graphs as the XAT-free chordal graphs. We provide alternative characterizations of these graphs, show that there are graph classes for which the various parameters differ by an arbitrary amount, and consider algorithms for computing catwidth.

Keywords: Graph algorithms; Treewidth; Pathwidth; Memory usage

1. Motivation and background

Dynamic programming algorithms on bounded treewidth graphs have been studied since the mid-1980s leading to several powerful approaches, see [7] for an overview. Recent efforts have been aimed at making these theoretically efficient algorithms amenable also to practical applications, for example, in the field of compiler optimization [12,19,6], and also as subroutines to solve planar graph problems [1–3]. It is clear that avoidance of time-consuming I/O to external memory becomes an important issue [5], and this forms the motivation for our study of the tradeoff between width and pathwidth of tree-decompositions. In a somewhat similar setting, the tradeoff between width and diameter of tree-decompositions has been studied, which has applications to dynamic and parallel algorithms [8]. We first give the background explaining how the speed of a bounded treewidth algorithm is connected to the pathwidth of the tree-decomposition used.

A tree-decomposition (T, X) of constant width k of a graph G shows, intuitively, that G can be constructed by gluing together graphs of size $k+1$, on vertex sets of size k, in a process guided by the tree T. A graph induced by a subtree of T will be a subgraph of G that is connected to the rest of G by at most k vertices. This explains why bottom-up dynamic programming on the tree-decomposition, that solves a problem brute-force on separators of constant size k and on gluings of such separators, can be used to efficiently solve a number of NP-hard graph problems on G. Such algorithms compute a table, of size exponential in k, for each node of the tree T, of partial solutions of the problem restricted to the subgraph of G induced by the tree rooted at this node. The large size of the tables means that these algorithms are memory-intensive. Since the information contained in the table at a child node of T is superfluous once the table of its parent has been updated, we look for a bottom-up traversal of T

☆ Part of this work done while visiting The Computer Science Department at ANU in Canberra, Australia.

E-mail address: telle@ii.uib.no (J.A. Telle).

0166-218X/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2004.01.012

that minimizes the number of tables that need to be stored simultaneously. In [5] a simple linear-time algorithm finding such a traversal for a tree T is given, and moreover it is shown that the minimum number of tables needed lies between the pathwidth of T and twice the pathwidth of T.

The natural issue that arises is therefore to study the tradeoff between the width k and the pathwidth of tree-decompositions, in the hope that I/O to external memory can be avoided or minimized. We will do this by studying, on the one hand, the catwidth parameter of G that reflects the lowest width obtainable when we restrict the tree T of the tree-decomposition to have pathwidth 1. Such trees are called caterpillars, and the catwidth of a graph lies between the treewidth and the pathwidth of the graph. See Fig. 1 for a very simple example showing three different tree-decompositions T, T', T'' of a graph and the number of tables in internal memory needed for each of them. We claim that the accepted observation that dynamic programming on a path-decomposition like T'' is easier than that on a general tree-decomposition, also holds for "caterpillar"-decompositions like T'. However, since both the size of tables in the tree-decomposition and the number of tables in memory (and thus, also the number of external memory references that may be necessary) may vary during computation, the current investigation is only a worst-case analysis. For the actual running time also further factors play a role, e.g. the time for updating a table based on another may also depend on the symmetric difference of nodes in the corresponding bags, and in this sense tree T'' in Fig. 1 is slightly worse than the other two. The actual effect of the current observations on the speed of treewidth algorithms must therefore be empirically tested, beyond that which was already reported in [5].

Just as treewidth relates to chordal graphs and pathwidth relates to interval graphs, catwidth relates to what we call catval graphs. A well-known concept in the study of graphs with a linear structure is that of an asteroidal triple (AT), see e.g. [10], and here we introduce extended asteroidal triples (XAT) to characterize catval graphs as exactly the XAT-free chordal graphs. The optimal situation may be said to arise when a graph has a tree-decomposition with pathwidth 1 which achieves the optimal width, i.e. when the treewidth of G is equal to the catwidth of G. It has been shown [17] that the multitolerance graphs have this property. On the other hand, it is clear that we must allow higher pathwidth, and that the width k is more important than the pathwidth, since the size of tables can be exponential in the width k while the number of tables needed is only linear in the pathwidth. We, therefore, also look at the spacewidth parameter of G, which accounts for this relative importance of width to pathwidth by multiplying the width of a tree-decomposition with the logarithm of its pathwidth.

We start by giving the basic definitions needed, and then look at alternative characterizations of the graphs having bounded value of the parameters. We show that there are graph classes, even trees, for which these parameters differ by an arbitrary amount. We also look at algorithms for these parameters, showing that it is NP-hard to compute the spacewidth or catwidth of a graph, but solvable in polynomial time to decide if these values are below a fixed constant k.

2. Basic definitions

We consider only connected, simple graphs with at least one edge. A leaf is a vertex with degree one. A tree is any graph that can be constructed by starting with a single vertex and iteratively adding leaves to the tree obtained so far. A path is a tree with two leaves. A caterpillar is a tree consisting of a path (the 'body') with added leaves (the 'hairs'). A graph G is a partial C-graph, for a class of graphs C, if G is a subgraph of a graph in C. A k-clique is a set of k vertices that induce a completely connected graph. A k-leaf is a vertex of degree k whose neighbors form a k-clique. A k-tree is any graph that can be constructed by starting with a k-clique and iteratively adding k-leaves to the k-tree obtained so far. A proper k-path is a k-tree with two k-leaves. A k-path is a k-tree consisting of a proper k-path with added k-leaves each of whose set of neighbors must form a separator of the proper k-path. Note that a 1-tree is a tree, a proper 1-path is a path, and a 1-path is a caterpillar but not necessarily a path. The above terminology for k-paths and proper k-paths has not always been fixed, but lately it seems to be what most authors have accepted as the standard.

Definition 1. A tree-decomposition (T, X) of a graph $G = (V(G), E(G))$ is a tree T and a collection X of subsets of vertices of G, called *bags*, with one bag $X_i \in X$ for each node $i \in V(T)$, such that for each edge $uv \in E(G)$ there is a node $i \in V(T)$ whose bag X_i contains both u and v, and for each vertex $v \in V(G)$ the nodes whose bag contains v form a connected subtree of T. The width of the tree-decomposition is the maximum number of vertices contained in any bag, minus one.

Definition 2. The treewidth $tw(G)$ of a graph G is the minimum width of any tree-decomposition (T, X) of G.

The pathwidth $pw(G)$ of a graph G is the minimum width of any tree-decomposition (T, X) of G where T is a path.

The fact that a graph G has treewidth at most k iff it is a partial k-tree, and pathwidth at most k iff it is a partial k-path explains why the k-path terminology has become the accepted one. We now define the main parameter studied in this paper.

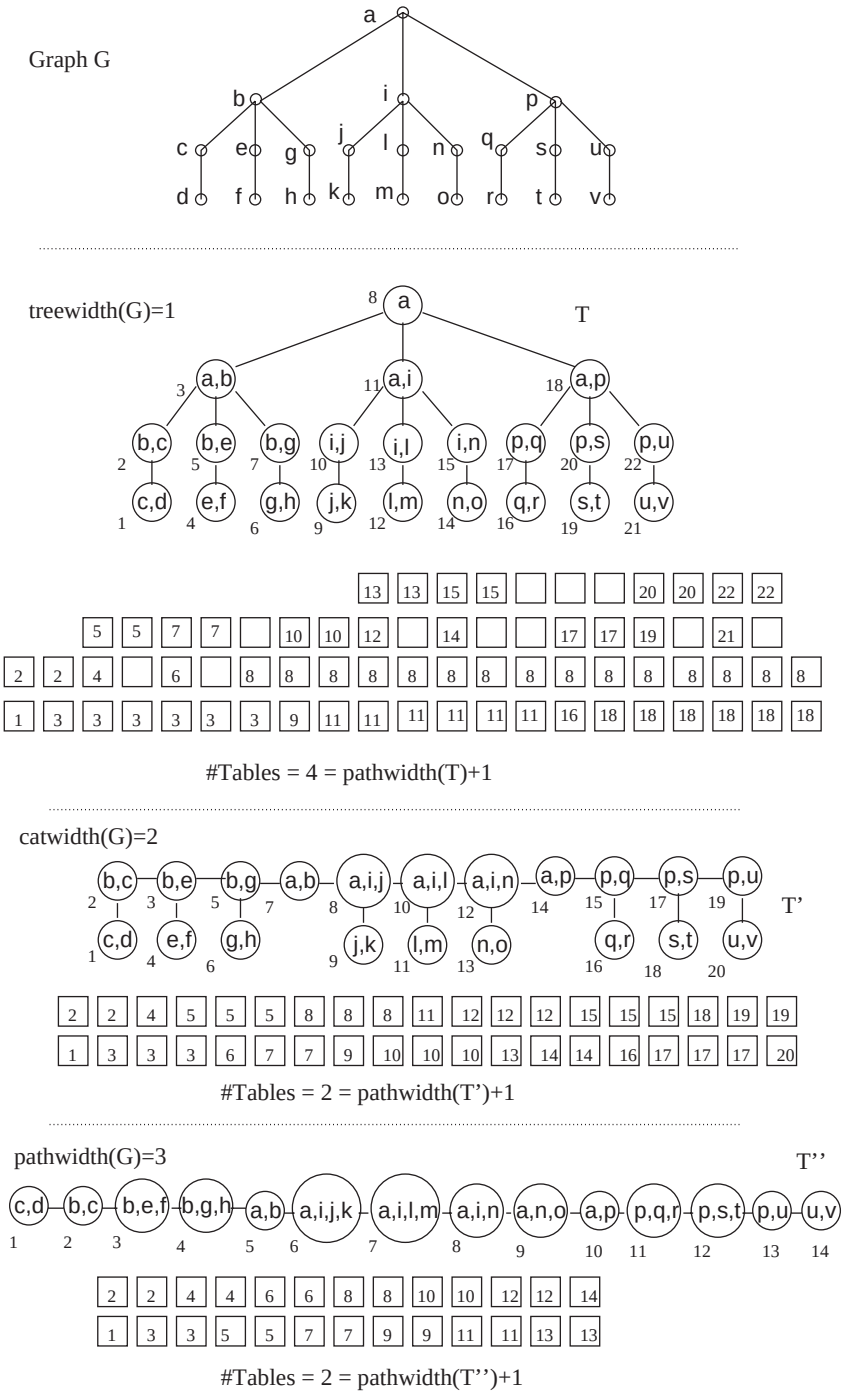

Fig. 1. Three tree-decompositions T, T', T'' of a graph G and the succession of tables stored in internal memory during dynamic programming on each of them.

Definition 3. The catwidth $catw(G)$ of a graph G is the minimum width of any tree-decomposition (T, X) of G where T is a caterpillar, i.e. where $pw(T) = 1$.

Since any path is a caterpillar and any caterpillar is a tree, we have $tw(G) \leqslant catw(G) \leqslant pw(G)$. In Fig. 1 is a graph with treewidth 1, catwidth 2 and pathwidth 3.

3. Catval graphs and *k*-caterpillars

The intersection graphs of subtrees of a tree, and subpaths of a path, are the well-known chordal graphs and interval graphs, respectively. We give an analogous definition for caterpillars.

Definition 4. We say that a graph G is a catval graph if G is the intersection graph of a set of connected subgraphs of a caterpillar.

It is well-known that graphs with treewidth (resp. pathwidth) at most k are exactly the subgraphs of chordal graphs (resp. interval graphs) of max clique size $k+1$. A similar result, with a straightforward proof, holds for catwidth.

Theorem 1. *A graph G has catwidth at most k iff G is the subgraph of a catval graph of max clique size $k+1$.*

Proof. Let G have a tree-decomposition (T, X) of width k with T a caterpillar. G has max clique size $k+1$ since for any clique of G there must be a bag containing those vertices. For a vertex v of G we define T_v to be the subgraph of T induced by the bags containing node v. By definition the graphs $\{T_v : v \in V(G)\}$ are connected subgraphs of the caterpillar T and it is easy to check that their intersection graph has max clique size $k+1$ and contains G. Conversely, from a set of connected subgraphs of a caterpillar T, we can construct a tree-decomposition (T, X) by taking as the bag for a node x of T precisely those vertices whose connected subgraph contains x. □

A well-known notion in the study of graphs with a linear structure is that of an asteroidal triple, which forms the basis of a famous result by Lekkerkerker and Boland from 1961 [15].

Definition 5. Three non-adjacent vertices x, y, z of a graph G form an AT if between any two of them there exists a path in G that avoids the neighborhood of the third.

Theorem 2 (*Lekkerkerker and Boland [15]*). *G is an interval graph iff it is chordal and AT-free.*

We give a similar result for catval graphs, based on the notion of an extended asteroidal triple. See Fig. 2 for an example.

Definition 6. An AT x, y, z of a graph G is an XAT if there exists vertices x', y', z' with $|\{x, y, z, x', y', z'\}| = 6$ such that $N[x'] \subset N[x]$, $N[y'] \subset N[y]$, $N[z'] \subset N[z]$. (where $\subset$ denotes strict inclusion).

Theorem 3. *G is a catval graph iff it is chordal and XAT-free.*

Proof. We will use the concept of a clique tree of a chordal graph G, which for our purposes can be viewed simply as a minimal tree-decomposition of G in which every bag induces a maximal clique. It is well-known that interval graphs are those chordal graphs having clique trees that form a path, and likewise it is easy to see that catval graphs are those chordal graphs having clique trees that form a caterpillar.

Assume G is chordal and has an XAT x, y, z. Since x, y, z is an AT, we know from Theorem 2 that any clique tree T of G has bags C, X, Y, Z such that $x, y, z \notin C$, $x \in X$, $y \in Y$, $z \in Z$ and removal of C from T disconnects T from X, Y, Z into three separate components. This holds since we have the three paths between any pair. Consider the one from x to y that avoids $N[z]$.

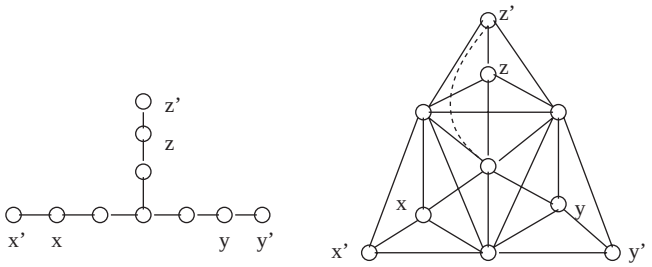

Fig. 2. Two chordal graphs that are not catval, as they have AT x, y, z that form an XAT. This is verified by x', y', z', whose closed neighborhoods are strictly contained in the closed neighborhoods of x, y, z, respectively. Addition of the dotted edge gives a catval graph.

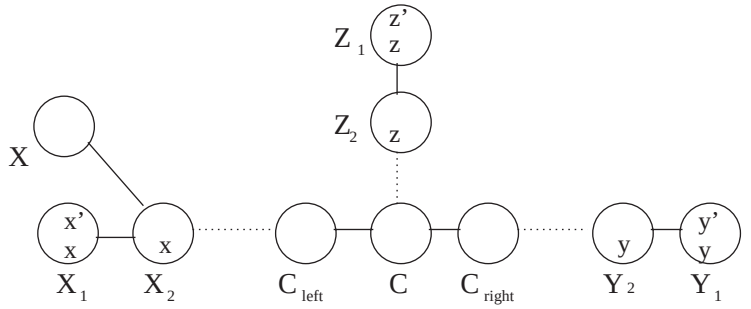

Fig. 3. Part of a clique tree with X_1, Y_1 endbags of the body.

This path must necessarily contain a vertex from each bag on the path in T from X to Y, and one of those bags, in particular C, cannot contain z, as every bag induces a clique. We claim that T cannot be a caterpillar, since if this were so we would have at least one of X, Y, Z, say Z, being a leaf of T with C its only neighbor. We know $z \in Z$ and $N[z'] \subset N[z]$ so there exists $w \in N(z)\backslash N(z')$, thus z' must belong to some bag Z' with $w \notin Z'$. But we cannot have $z' \in C$, since C contains at least one vertex not in $N[z]$, namely the vertex on the x, y-path avoiding $N[z]$. Thus Z is not a leaf, T is not a caterpillar, and G is not a catval graph.

For the other direction of the proof, let G be chordal and XAT-free. We will show that it has a clique tree which is a caterpillar, i.e. a path (the body) with added leaves (the hairs). For a given tree T with two chosen leaves X and Y we assign a quadruple of integers (M, N, D, S) that will indicate how close the tree is to being an 'optimal' caterpillar with body the path between X and Y. Let D be the length of the path between X and Y, let M be the maximum distance of any leaf to this body, let there be N leaves at this distance M, and let $S = |N(X')| + |N(Y')|$ for the unique neighbors X', Y' of the leaves X, Y. For the tree T we assign the lexicographically smallest quadruple (in left-to-right decreasing order of importance) (M, N, D, S) thus assigned over all choices of X and Y. Note that $M \leqslant 1$ iff T is a caterpillar. Now, over all clique trees of G, pick one, call it T, that has the lexicographically smallest quadruple thus assigned, say (M_T, N_T, D_T, S_T). Recall 1. M_T: Max distance of leaf to body, 2. N_T: Number of leaves at max distance, 3. D_T: Length of body, 4. S_T: Size of neighborhood of neighbors of endvertices of body. We will show that if $M_T \geqslant 2$ we can find a clique tree T' of G such that its quadruple $(M_{T'}, N_{T'}, D_{T'}, S_{T'})$ is lexicographically smaller than (M_T, N_T, D_T, S_T), in contradiction to the minimal choice of T.

If $M_T \geqslant 2$ we can find bags C, X, Y, Z in T such that X and Y are the endbags of the chosen body, C is a 'central' bag on the body, Z is a leaf bag at maximum distance from the body, and removal of C from T disconnects T from X, Y, Z into three separate components. Let the path from X to C be $X = X_1, X_2, \ldots, X_{j-1} = C_{\text{left}}, X_j = C$ for some $j \geqslant 3$. See Fig. 3. Note that we must have a vertex $x' \in X_1$ and $x' \notin X_2$ since otherwise $X_1 \subseteq X_2$ and T is not a clique tree since either $X_1 = X_2$ or X_1 does not induce a maximal clique. Likewise, we must have a vertex $x \in X_1, x \in X_2, x \notin X_3$ since otherwise $X_1 \cap X_2 \subseteq X_3$ and the tree T' obtained by dropping the (X_1, X_2)-edge and instead making X_1 adjacent to X_3 would be a lexicographically lower clique tree of G. We distinguish two cases in the argument for this latter point.

Case 1: $N(X_2) = \{X_1, X_3\}$. Endbags X_1, Y gives a body for T' with shorter body, i.e. where $M_{T'} = M_T, N_{T'} = N_T$ and $D_{T'} < D_T$.

Case 2: We have some $\bar{X} \in N(X_2)\backslash\{X_1, X_3\}$. Endbags $\bar{X}, Y$ gives a body for T' with smaller neighborhood for the neighbors of the endbags, i.e. where $M_{T'} = M_T, N_{T'} = N_T, D_{T'} = D_T$ and $S_{T'} < S_T$.

A similar argument identifies vertices y' and y at the other end Y of the body. Let the path from Z to C be $Z = Z_1, Z_2, \ldots, Z_i = C, i \geqslant 3$. Note that we must have a vertex $z' \in Z_1$ and $z' \notin Z_2$ since otherwise $Z_1 \subseteq Z_2$ and T is not a clique tree since either $Z_1 = Z_2$ or Z_1 does not induce a maximal clique. Likewise, we must have a vertex $z \in Z_1, z \in Z_2, z \notin Z_3$ since otherwise $Z_1 \cap Z_2 \subseteq Z_3$ and the tree T' obtained by dropping the (Z_1, Z_2)-edge and instead making Z_1 adjacent to Z_3 would be a clique tree of G where the same choice of endbags would give lower maximum distance to a leaf or fewer vertices at maximum distance, i.e. with $M_{T'} < M_T$ or with$M_{T'} = M_T$ and $N_{T'} < N_T$.

Now, we will argue that x, y, z is an XAT in G. We have already shown that there exists x', y', z' such that $N[x'] \subset N[x], N[y'] \subset N[y], N[z'] \subset N[z]$. It remains to show that there exists a path between any two that avoids the neighborhood of the third. We construct such an x, y-path by choosing (in such a way that the path becomes simple) from each bag on the X, Y-path a vertex that is not in Z_2, and hence not in $N[z]$. Such a vertex must exist in each of these bags, and we show this in particular for the bags $C_{\text{left}}, C, C_{\text{right}}$ where (C_{left}, C) and (C, C_{right}) are assumed to be edges of the X, Y-path. If there does not exist a vertex $a \in C_{\text{left}} \cap C$ with $a \notin Z_2$, then $C_{\text{left}} \cap C \subseteq Z_2$ and the tree T' obtained by removing the edge (C_{left}, C) and adding the edge (C_{left}, Z_2) would be a clique tree of G where the same choice of endbags would give lower maximum distance

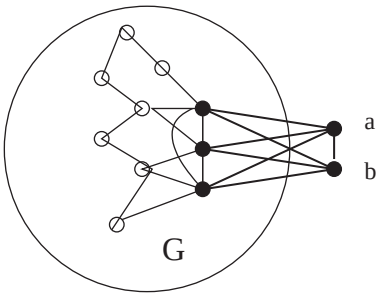

Fig. 4. The black vertices form a (3,2)-flap, and its addition to G introduced the new vertices a and b and the thick edges, each incident to at least one new vertex.

to a leaf or fewer vertices at maximum distance, i.e. with $M_{T'} < M_T$ or with $M_{T'} = M_T$ and $N_{T'} < N_T$. Likewise, there must exist a vertex $b \in C \cap C_{\text{right}}$ with $b \notin Z_2$ (note that we may have $a = b$). Thus, we can choose a and b to be consecutive vertices on the x, y-path. The arguments that there must exist an x, z-path that avoids the neighborhood of y and a z, y-path that avoids the neighborhood of x, are similar. We give the argument for the x, z-path, which is constructed by choosing from each bag on theX, Z-path a vertex that is not in Y_2. If such a vertex does not exist in, say $C_{\text{left}} \cap C$, then $C_{\text{left}} \cap C \subseteq Y_2$ and the tree T' obtained by removing the edge (C_{left}, C) and adding the edge (C_{left}, Y_2) would be a clique tree of G where endbags X, Z would give the body $X, \ldots, C_{\text{left}}, Y_2, Y_3, \ldots, C_{\text{right}}, C, \ldots, Z$ having lower maximum distance to a leaf or fewer vertices at maximum distance, i.e. with $M_{T'} < M_T$ or with $M_{T'} = M_T$ and $N_{T'} < N_T$. This follows since the only new leaf is Y which is at distance 1, and the old maximum distance leaf Z is now on the body. Similarly, we can find a vertex in $C \cup Z_{i-1}$ not in Y_2 since otherwise the tree T' obtained by removing edge (C, Z_{i-1}) and adding edge (Z_{i-1}, Y_2) is a clique tree where the choice of body with endpoints X, Z has lower maximum distance or fewer vertices at maximum distance.

We conclude that the assumption that $M_T \geqslant 2$ leads to the existence of an XAT in G, a contradiction. Thus, $M_T < 2$, T is a caterpillar and G is a catval graph. □

Our next goal is to define k-caterpillars, so that k-caterpillars will relate to catwidth k in the same way that k-trees relate to treewidth k and k-paths to pathwidth k. We first generalize the notion of a k-leaf. See Fig. 4 . Note that a k-leaf is a $(k, 1)$-flap.

Definition 7. For $p, q \geqslant 1$, a (p, q)-flap in a graph G consists of taking two disjoint sets P and Q of, respectively, p and q vertices, that together induce a $(p + q)$-clique, such that P separates Q from the rest of the graph. The operation of adding a (p, q)-flap to a graph G consists in taking for P an existing p-clique in G, and adding the q new vertices Q, together with the new edges incident to the vertices of Q.

Definition 8. A k-caterpillar is a k-tree consisting of a k-path with added (p, q)-flaps, for any $p + q = k + 1$.

Note that a 1-caterpillar is a caterpillar with added leaves, just as a 1-path is a path with added leaves.

Theorem 4. *A graph G has catwidth at most k iff G is a partial k-caterpillar.*

Proof. For one direction: Let G be a subgraph of a k-caterpillar A that consists of a k-path B with added (p, q)-flaps. It is well-known that $pw(B) = k$, and we construct a tree-decomposition of A, and hence also of G, starting from an optimal path-decomposition (T, X) of B. Let (P, Q) be a (p, q)-flap of A that was added to B with $|P| = p$, $|Q| = q$, $p + q = k + 1$. Since P is a clique of B we have $P \subseteq X_i$ for some bag X_i of (T, X). Make a new bag containing the vertices $P \cup Q$ and make this bag adjacent to X_i. Doing this for each (p, q)-flap of A that was added to B we end up with the desired tree-decomposition (T', X') of width $k + 1$ of G with T' a caterpillar. For the other direction: Let (T', X') be a tree-decomposition of width $k + 1$ of G with T'a caterpillar consisting of a path T with added leaves. It is well-known that the path-decomposition induced by T is the path-decomposition of a subgraph C of G to which we can add edges and get a k-path B s.t. every bag of T induces a clique in B. Let S be a bag of $T' - T$, adjacent to bag X_i of T. We know there is a $(k + 1)$-clique K of the k-path B with $X_i \subseteq K$. We add a (P, Q)-flap to B where $Q = S - X_i$ and $P = S \cap X_i \cup R$, where the vertices R are chosen arbitrarily from $K - X_i$ so that $|P| = k + 1 - |Q|$. Doing this for all bags of $T' - T$ constructs a k-caterpillar having G as a subgraph. □

4. Graphs with extreme parameter values

As explained in the introduction, from a practical viewpoint we must allow the tree in the tree-decomposition to have higher pathwidth than 1. Moreover, for memory usage the width k is more important than the pathwidth of the tree, since the size of tables is exponential in the width k while the number of tables needed is only linear in the pathwidth. The spacewidth parameter of G is an attempt to model this relative importance.

Definition 9. The spacewidth $spacew(G)$ of a graph G is the minimum, over all tree-decompositions (T, X) of G, of the product of the logarithm of the pathwidth of the tree T plus one, and the width of the tree-decomposition, i.e.

$$spacew(G) = \min_{(T,X)} \{\lceil \log_2(pw(T)+1)\rceil (\max_{X_i \in X} \{|X_i|\} - 1)\}.$$

Note that the factor $\lceil \log_2(pw(T)+1)\rceil$ is 1 only for $pw(T) = 1$. Since no tree has pathwidth less than 1, and the pathwidth of a caterpillar is 1, and any path is a caterpillar, we have the spacewidth parameter lying between the treewidth and catwidth parameters: $tw(G) \leqslant spacew(G) \leqslant catw(G) \leqslant pw(G)$. In the remainder of this section we prove the following:

Theorem 5. *There are trees with arbitrarily large spacewidth, and the difference between the catwidth and pathwidth parameters can be arbitrarily large, also for trees.*

Allowing increased pathwidth of the tree-decomposition can affect the width substantially. For example, the 2-tree G in Fig. 5 has a tree-decomposition (T, X) of width 2 with $pw(T) = 2$, while any tree-decomposition (T', X') with $pw(T') = 1$ has width at least 5, thus $spacew(G) = 4 < catw(G)$. We can show this by letting T be the tree underlying the 2-tree construction process, i.e. a 6-star with each edge subdivided 5 times, and naturally letting each triangle of the 2-tree be a bag of size 3. However, for (T', X') with $pw(T') = 1$, all 3 vertices in the center triangle of the 2-tree must belong to at least one bag in the 'induced' path-decomposition of one of the 6 proper 2-path 'tentacles'. At least 3 vertices from this proper 2-path, apart from the center triangle vertices, must also be in that bag, for a minimum of 6 vertices in some bag.

We next show that the difference between the catwidth and pathwidth parameters can be arbitrarily large, even for trees.

Definition 10. Let T_i be the complete ternary tree with height $i+1$, defined as follows: T_0 is a tree with a single node, which is also its root, and T_i, $i \geqslant 1$ is a tree with a root having three children such that the subtree rooted at each child is a copy of T_{i-1}.

Lemma 1. *The difference between the catwidth and pathwidth of the complete ternary tree $T_{f(j)+j}$ is at least $j+1$, for any $j \geqslant 0$ and $f(j) = 1 + (3^j - 1)/2$.*

Proof. It is well-known that $pw(T_i) = i$ for $i \geqslant 1$ [11]. The subtrees rooted at vertices of height $j \leqslant i$ in T_i are copies of T_j. Consider a path-decomposition (P, X) of $T_{f(j)-1}$ of width $f(j)-1$, for any $j \geqslant 0$. Starting from this $T_{f(j)-1}$ we can construct a copy of $T_{f(j)+j}$ by adding, to each leaf x of $T_{f(j)-1}$, three copies of T_j. Likewise, starting from (P, X) we can construct a tree-decomposition (T, X) of $T_{f(j)+j}$ with T a caterpillar, by adding a leaf to P for each added copy of T_j, adjacent to a vertex

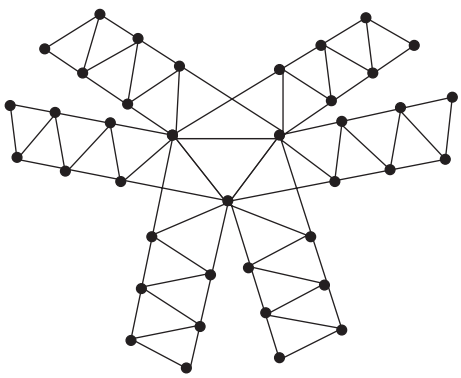

Fig. 5. A 2-tree with a tree-decomposition of width 2 and pathwidth 2, for which every tree-decomposition with pathwidth 1 has width at least 5.

in P whose bag contains the leaf x to which this copy was added, with the vertices in this new leaf bag being those of T_j and x. The size of this bag is $|V(T_j)| + 1 = 1 + (3^j - 1)/2 = f(j)$. The width of (T, X) is the same as the width of (P, X), namely $f(j) - 1$, and we thus have $catw(T_{f(j)+j}) \leqslant f(j) - 1$ and $pw(T_{f(j)+j}) = f(j) + j$. □

Even if all trees have treewidth 1, there exist trees with arbitrarily large pathwidth, like the complete ternary trees, so it is an obvious question if there are also trees with arbitrarily large spacewidth. The following shows that the answer to this question is affirmative.

Lemma 2. *If a connected graph G has a tree-decomposition (T, X) of width k, then $pw(T) \geqslant (pw(G) - k)/(k + 1)$.*

Proof. Assume T has a path-decomposition (P, X') of width $pw(T)$. We can then construct a path-decomposition (P, X'') of G by 'expanding' the bags of P to contain the actual vertices of G that are members of corresponding bags of T. The pathwidth of G is thus at most the width of (P, X'') which is at most $(k + 1)(pw(T) + 1) - 1$. □

5. Algorithms for catwidth and spacewidth

For a cograph G, its catwidth and spacewidth together with decompositions can be computed in linear time. This by the algorithm given in [9] to compute the pathwidth and a corresponding path-decomposition of a cograph, since it is also shown that treewidth = pathwidth for cographs, and therefore also pathwidth = catwidth = spacewidth = treewidth.

Just as for graphs of bounded treewidth and pathwidth, it is easy to see that the graphs of bounded catwidth and spacewidth are closed under minors.

Fact 1. *For any fixed $k \geqslant 1$ the classes of graphs $\{G : spacew(G) \leqslant k\}$ and $\{G : catw(G) \leqslant k\}$ are closed under minors.*

It follows from the work of Robertson and Seymour [18] that for each of these graph classes there exists a finite list of minimal forbidden minors (note their proof is not constructive) and thus polynomial time algorithms to decide membership, for fixed k. Since a bound on these parameters implies a bound also on the treewidth, these algorithms are linear-time.

Corollary 3. *For any fixed $k \geqslant 1$, the problem of deciding if a given graph on n vertices belongs to $\{G : spacew(G) \leqslant k\}$ or to $\{G : catw(G) \leqslant k\}$ is solvable in $O(n)$ time.*

The minimal forbidden minors for the class of graphs of catwidth 1 are the triangle and the 10-vertex graph arising from subdividing each edge of a 3-star 2 times. For graphs of catwidth 2 we may guess that there are many minimal forbidden minors, just as for pathwidth 2 where there are over 100 [14]. Deciding if a tree T has catwidth (and hence also spacewidth) 1 is easily done in linear time, but for higher values of these parameters giving an explicit such algorithm is left for future work. For an AT-free graph G, we have $tw(G) = pw(G)$ [16], so also $tw(G) = spacew(G) = catw(G) = pw(G)$. In [4] it is shown that determining treewidth is NP-hard for co-comparability graphs (although the authors did not use this terminology for their graph class). Since co-comparability graphs are AT-free and thus have treewidth equal to pathwidth, we have:

Corollary 4. *Determining the catwidth or spacewidth of a cocomparability graph is NP-hard.*

6. Conclusion

Based on the observation that algorithms on graphs of small treewidth are faster if tables fit into memory, we have introduced two parameters related to pathwidth and treewidth, called catwidth and spacewidth. The catwidth parameter led naturally to the definition of the class of catval graphs, and to the XAT.

Various further questions related to these notions should be studied. A very recent result [13] gives a linear-time algorithm for recognizing catval graphs. On the practical side we have already in the introduction mentioned that experimentation should be carried out in order to evaluate the importance of these observations. Various other issues, in analogy with results on pathwidth and treewidth, as well as interval and chordal graphs, and also AT, are also open for further investigations.

References

[1] J. Alber, H.L. Bodlaender, H. Fernau, R. Niedermeier, Fixed parameter algorithms for planar dominating set and related problems, Algorithmica, 33 (2002) 461–493.
[2] J. Alber, H. Fernau, R. Niedermeier, Parameterized complexity: exponential speed-up for planar graph problems, Proceedings ICALP 2001, Crete, Greece, Lecture Notes in Computer Science, vol. 2076, Springer, Berlin, 2001, pp. 261–272.
[3] J. Alber, R. Niedermeier, Improved tree decomposition based algorithms for domination-like problems, in: Proceedings LATIN 2002. Cancun, Mexico, April 2002, Lecture Notes in Computer Science, Springer, Berlin, 2001.
[4] S. Arnborg, D. Corneil, A. Proskurowski, Complexity of finding embeddings in a k-tree, SIAM J. Alg. Discr. Meth. 8 (1987) 277–284.
[5] B. Aspvall, A. Proskurowski, J.A. Telle, in: H. Bodlaender (Ed.), Memory requirements for table computations in partial k-tree algorithms, Algorithmica 27(3) (2000) 382–394 (Special issue on Treewidth, Graph Minors and Algorithms).
[6] H. Bodlaender, J. Gustedt, J.A. Telle, Linear-time register allocation for a fixed number of registers, in: Proceedings of the SODA'98, San Fransisco, USA, 1998, pp. 574–583.
[7] H. Bodlaender, Treewidth: algorithmic techniques and results, Proceedings MFCS'97, Lecture Notes in Computer Science, vol. 1295, Springer, Berlin, 1997, pp. 29–36.
[8] H. Bodlaender, T. Hagerup, Tree decompositions of small diameter, Proceedings MFCS'98, Lecture Notes in Computer Science, vol. 1450, Springer, Berlin, 1998, pp. 702–712.
[9] H. Bodlaender, R. Möhring, The pathwidth and treewidth of cographs, SIAM J. Discr. Math. 6 (2) (1993) 181–188.
[10] D. Corneil, S. Olariu, L. Stewart, Asteroidal triple-free graphs, SIAM J. Discr. Math. 10 (3) (1997) 399–430.
[11] J.A. Ellis, I.H. Sudborough, J.S. Turner, The vertex separation number and search number of a graph, Inform. Comput. 113 (1994) 50–79.
[12] J. Gustedt, O. Mæhle, J.A. Telle, The treewidth of Java programs, Proceedings ALENEX'02-Fourth Workshop on Algorithm Engineering and Experiments, San Francisco, January 4–5, 2002, Lecture Notes in Computer Science, Springer, Berlin, vol. 2409.
[13] M. Habib, C. Paul, J.A. Telle, A linear-time algorithm for recognition of catval graphs, Montpellier TR-LIRMM 03-004, March 2003.
[14] N.G. Kinnersley, M.A. Langston, Obstruction set isolation for the Gate matrix layout problem, Discr. Appl. Math. 54 (1994) 169–213.
[15] C.G. Lekkerkerker, J.Ch. Boland, Representation of a finite graph by a set of intervals on the real line, Fund. Math. 51 (1962) 45–64.
[16] R. Möhring, Triangulating graphs without asteroidal triples, Discr. Appl. Math. 64 (1996) 281–287.
[17] A. Parra, Triangulating multitolerance graphs, Discr. Appl. Math. 84 (1998) 183–197.
[18] N. Robertson, P. Seymour, Graph minors I. Excluding a forest, J. Combin. Theory B 35 (1983) 39–61.
[19] M. Thorup, Structured programs have small tree-width and good register allocation, Inform. Comput. 142 (1998) 159–181.